UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/729,390	12/04/2000	Koji Takahara	0828.64986	6429

24978 7590 02/25/2004

GREER, BURNS & CRAIN
300 S WACKER DR
25TH FLOOR
CHICAGO, IL 60606

EXAMINER	
ROCHE, TRENTON J	

ART UNIT	PAPER NUMBER
2124	6

DATE MAILED: 02/25/2004

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	09/729,390	TAKAHARA ET AL.	
	Examiner	Art Unit	
	Trent J Roche	2124	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *19 December 2003*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-12* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-9,11 and 12* is/are rejected.

7)☒ Claim(s) *10* is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *19 December 2003* is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☒ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☒ All b)☐ Some * c)☐ None of:

1.☒ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☐ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

DETAILED ACTION

1. This office action is responsive to Amendment A filed 19 December 2003.

2. Per Applicant's request, claim 2 has been amended. Claims 1-12 are pending in the application.

3. Upon further review of the application in light of the Applicant's arguments, the rejection of claim 10 under 35 U.S.C. 103(a) made in the earlier office action is withdrawn.

Priority

4. Acknowledgment is made of applicant's claim for foreign priority under 35 U.S.C. 119(a)-(d). The certified copy has been filed in parent Application No. 09/729,390, filed on December 4, 2000.

Claim Rejections - 35 USC § 102

5. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (a) the invention was known or used by others in this country, or patented or described in a printed publication in this or a foreign country, before the invention thereof by the applicant for a patent.

6. Claims 1-5, 8 and 11 are rejected under 35 U.S.C. 102(a) as being anticipated by U.S. Patent 6,110,227 to Marcelais et al.

Regarding claim 1:

Marcelais et al teach:

- an information processing apparatus ("a conventional personal computer...including a processing unit..." in col. 5 lines 48-49)

- translating a source file including a dynamic variable into an object file by a compiling process and converting the object file into an executable load module by a linking process ("A compiler...then converts the source code...into machine-readable object code, which is stored in one or more files...A linker...processes a file to produce a binary image..." in col. 4 lines 30-36)
- dynamic variable specifying means for specifying a target dynamic variable from the source file ("the variable being identified by a variable symbol in the file..." in col. 18 lines 43-44)
- area specifying means for specifying areas ensured in the case of a dynamic variable specified by the dynamic variable specifying means being developed into a memory at the time of executing the load module ("in order to write the initialized value into the object file, a new section is created...in the object file to store the initialized value of the variable..." in col. 14 lines 8-12)
- initializing means for initializing areas specified by the area specifying means to a predetermined value ("an initializer is processed...to assign a default value of a variable." in col. 3 lines 45-47)

Regarding claim 2:

The rejection of claim 1 is incorporated, and further, Marcelais et al teach an area integrating means for integrating specified areas ensured in the case of a plurality of different dynamic variables specified by the dynamic variable specifying means being developed into the memory. ("an uninitialized data area is created in the temporary memory of the computer." in col. 3 lines 54-55)

Regarding claim 3:

The rejection of claim 1 is incorporated, and further, Marcelais et al teach a variable integrating means for integrating dynamic variables dispersed in one or more object files integrated by the linking process ("A 'symbol table' is broadly defined as a list of...variables, routines, and so on. A symbol table is stored as part of the object file...so that the linker can resolve references between sections of the object file and separately compiled modules or files." in col. 4 lines 45-51)

Regarding claim 4:

The rejection of claim 1 is incorporated, and further, Marcelais et al teach allocating a target dynamic variable to a new data section ("an uninitialized data area is created in the temporary memory of the computer." in col. 3 lines 54-55) wherein an initialization means is performed on a dynamic variable in the data section ("This temporary memory is then initialized by the appropriate initializers." in col. 3 lines 55-56)

Regarding claim 5:

The rejection of claim 4 is incorporated, and further, Marcelais et al teach when a plurality of object files are linked to generate a load module the initializing means does not perform an initializing process on a dynamic variable ("A linker...processes a file to produce a binary image...a linker involves linking object files together..." in col. 4 lines 35-37), and further, ("When a...compiler encounters...an uninitialized global variable...the compiler generates an additional function called an 'initializer,' that will be responsible for initializing the global variable upon start-up of the binary image." in col. 4 lines 55-60. The initialization takes place when the program is run, not when the object files are linked.)

Regarding claim 8:

The rejection of claim 1 is incorporated, and further, Marcelais et al teach an initialized variable specifying means for specifying a dynamic variable on which the initializing process is performed ("When a...compiler encounters...an uninitialized global variable...the compiler generates an additional function called an 'initializer,' that will be responsible for initializing the global variable upon start-up of the binary image." in col. 4 lines 55-60)

Regarding claim 11:

Marcelais et al teach:

- a computer-readable record medium recording a computer program ("a computer readable medium having stored thereon computer-executable instructions..." in col. 19 lines 42-43)
- translating a source file including a dynamic variable into an object file by a compiling process and converting the object file into an executable load module by a linking process ("A compiler...then converts the source code...into machine-readable object code, which is stored in one or more files...A linker...processes a file to produce a binary image..." in col. 4 lines 30-36)
- dynamic variable specifying means for specifying a target dynamic variable from the source file ("the variable being identified by a variable symbol in the file..." in col. 18 lines 43-44)
- area specifying means for specifying areas ensured in the case of a dynamic variable specified by the dynamic variable specifying means being developed into a memory at the time of executing the load module ("in order to write the initialized value into the object file, a new section is created...in the object file to store the initialized value of the variable..." in col. 14 lines 8-12)

- initializing means for initializing areas specified by the area specifying means to a predetermined value ("an initializer is processed...to assign a default value of a variable." in col. 3 lines 45-47)

Claim Rejections - 35 USC § 103

7. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

8. Claims 6 and 7 are rejected under 35 U.S.C. 103(a) as being unpatentable over U.S. Patent 6,110,227 to Marcelais et al in view of U.S. Patent 6,219,834 to Soroker et al.

Regarding claim 6:

The rejection of claim 1 is incorporated, and further, Marcelais et al do not disclose an initial-value entering means used by the initializer before a compiling process. Soroker et al disclose in an analogous compiler system an initial-value entering means used by the initializer before a compiling process ("the...compiler system receives an input program...which contain...at least one configuration file." in col. 3 lines 6-8 and further, "The configuration file includes such information as...variable assignments..." in col. 4 lines 4-8. The initial values are stored in the configuration file, which exists before a compilation process). It would have been obvious to someone of ordinary skill in the art at the time the invention was made to use the variable initialization process of Soroker et al

with the pre-processing system of Marcelais et al, as this would give a developer greater control in debugging a computer program in the system disclosed by Marcelais et al, as the developer would be able to specify initialization data.

Regarding claim 7:

The rejection of claim 1 is incorporated, and further, Marcelais et al disclose an initialization means before execution of the load module ("an initializer is pre-processed after compilation of the source code, but prior to linking of object and/or library files." in col. 7 lines 18-20). Marcelais et al do not disclose an initial-value entering means. Soroker et al disclose an initial-value entering means. Note the rejection regarding claim 6.

9. Claim 9 is rejected under 35 U.S.C. 103(a) as being unpatentable over U.S. Patent 6,110,227 to Marcelais et al in view of U.S. Patent 6,523,097 to Liedtke et al.

Regarding claim 9:

The rejection of claim 1 is incorporated, and further, Marcelais et al do not disclose error informing means for informing of an error in the case of a dynamic variable which holds an initial value being referred to at the time of execution of a load module. Liedtke et al disclose in an analogous variable initialization system an error informing means for informing of an error in the case of a dynamic variable which holds an initial value being referred to at the time of execution of a load module ("For detecting uninitialized variables, a computer processing system initializes all variables upon creation with the unvalued. The reading of such a variable prior to writing a value into it raises an exception." in col. 5 lines 22-25) It would have been obvious to someone of ordinary skill in the art

at the time the invention was made to use the error informing method of Liedtke et al in the variable initialization system of Marcelais et al, as this would alert the developer of existing errors in the program code which could cause undesirable program operation in the computer system disclosed by Marcelais et al.

10. Claim 12 is rejected under 35 U.S.C. 103(a) as being unpatentable over U.S. Patent 6,110,227 to Marcelais et al in view of U.S. Patent 6,085,029 to Kolawa et al.

Regarding claim 12:

Marcelais et al teach:

- an information processing apparatus ("a conventional personal computer...including a processing unit..." in col. 5 lines 48-49)
- translating a source file into an object file by a compiling process and converting the object file into an executable load module by a linking process ("A compiler...then converts the source code...into machine-readable object code, which is stored in one or more files...A linker...processes a file to produce a binary image..." in col. 4 lines 30-36)
- specifying means for specifying a target from the source file ("being identified by a variable symbol in the file..." in col. 18 lines 43-44)
- initializing means for initializing areas ensured by the area ensuring means to a predetermined value ("all global variables...are set to a default value upon start-up..." in col. 4 lines 3-4)

Marcelais et al do not disclose an area ensuring means for ensuring, at the time of executing the load module, areas in a memory being a predetermined number of bytes more than areas declared in an

array specified by the array specifying means, nor discloses the use of an array. Kolawa et al disclose in an analogous memory management computer system a memory ensuring means for arrays as claimed ("For arrays, the array variable and its size must be declared to the error checking engine. For each write operation to that array, the error-checking engine must check if the index into the array is valid." in col. 9 line 65 to col. 10 line 2). It would have been obvious to someone of ordinary skill in the art at the time the invention was made to use the memory ensuring means of Kolawa et al with the variable initialization system of Marcelais et al, as this would ensure that the variables being initialized are all within available memory space of the system disclosed by Marcelais et al.

Allowable Subject Matter

11. Claim 10 is objected to as being dependent upon a rejected base claim, but would be allowable if rewritten in independent form including all of the limitations of the base claim and any intervening claims.

12. The following is a statement of reasons for the indication of allowable subject matter: Claim 10, dependent on claim 9, further dependent on claim 1, discloses a dynamic variable specifying means, a means for specifying an area for a dynamic variable, and more specifically, *a means for ensuring that a memory area being a predetermined number more than the number of elements of an array declared in source code if the dynamic variable is an array, wherein the initializing means initializes all areas in a memory ensured by the memory ensuring means to a predetermined value, further wherein the error informing means informs of an error in the case of an array which holds the initial value being referred to.* The prior art of record, specifically Marcelais et al, taken alone or in combination with Kolawa et al, do not disclose a determination of whether a dynamic variable is an array, and if so, ensuring that a memory area is a predetermined

number more than the elements of an array if the dynamic variable is an array, wherein the initializing means initializes all areas in a memory ensured by the memory ensuring means to a predetermined value, further wherein the error informing means informs of an error in the case of an array which holds the initial value being referred to.

Instead, Marcelais et al discloses a determination of whether a dynamic variable exists, an area specifying means for the dynamic variable, and an initialization means for the dynamic variable prior to execution of the program. Marcelais et al do not disclose a dynamic variable being an array, and an ensuring means for ensuring that the memory area for the dynamic variable array is of sufficient size. Kolawa et al discloses a memory management and debugger wherein a means for ensuring that the memory area for an array is of sufficient size, however, Kolawa et al does not suggest or teach a determination of whether a dynamic variable represents an array, nor an error information means in the case of an array which holds an initial value being referred to.

Response to Arguments

13. Applicant's arguments filed 19 December 2003 have been fully considered but they are not persuasive.

Per claims 1-5, 8 and 11:

The Applicant states that Marcelais et al does not teach or suggest the features of the dynamic variable specifying means or the area specifying means described in accordance with claim 1. However, the Applicant merely states that Marcelais et al relates to a system and method for pre-processing variable initializers, and does not clearly point any specific error in the rejection set forth in the previous office action. Moreover, Marcelais et al does indeed recognize a problem associated

with the initialization of a dynamic variable during program execution ("a common problem associated with large C++ programs is a significant latency in start-up time due to the initialization of global variables" in col. 1 lines 30-32). The global variables discussed in Marcelais et al are variables which are initialized during execution, which is interpreted to represent a dynamic variable. A dynamic variable, as stated in the Applicant's remarks, is a variable wherein allocation "to a memory area is determined when a program is executed." (page 10 of the remarks). A dynamic variable specifying means is also disclosed in Marcelais et al, as shown in col. 4 lines 55-60 ("When a C++ compiler encounters in the source code an uninitialized global variable...to require initialization, the compiler generates an additional function called an 'initializer'...). Finally, Marcelais et al also disclose an area specifying means for a dynamic variable being developed into the memory at the time of execution ("In order to write the initialized value into the object file, a new section is created...in the object file to store the initialized value of the variable..." in col. 14 lines 8-12). Therefore, the rejections of claim 1-5, 8 and 11 are proper and herein maintained.

Per claims 6, 7 and 9:

The Applicant states that claims 6, 7 and 9 are allowable for being dependent on claims 1. However, the rejection of claim 1 was shown to be proper, and as such, the rejections of claim 6, 7 and 9 are proper.

Per claim 12:

The Applicant states that the combination of Marcelais et al in view of Kolawa et al does not teach or suggest any means for ensuring areas in the memory which is a predetermined number of bytes more than areas declared in a specified array. The Applicant further states that detecting an array

operation that attempts to access an invalid memory location is not the same as ensuring that there are more areas in the memory than that which is declared in the specified array. In response, the Examiner notes that Kolawa et al disclose the definition of a memory area in col. 10 lines 4-8, ("the engine must perform a test for whether the memory address being written to falls outside of a valid memory address range defined by the dimension operand used to declare the memory block size..."). The area ensuring means as recited, given the broadest reasonable interpretation, performs the same function as what is stated in Kolawa et al. The Applicant's claim has the ability to ensure that the number of memory areas is greater than the object that will be assigned to them, and Kolawa et al's invention discloses a test to ensure that the number of memory areas is not less, and therefore, greater or equal to the object that will be assigned to them by checking to see if the intended address falls beyond the defined dimensional bounds of the memory block.

The Applicant further states that there is not suggestion or teaching to combine the references of Marcelais et al with Kolawa et al. However, it is noted that both Marcelais et al and Kolawa et al, while being aimed at solving different problems, both are analogous in that they deal with dynamic memory management and problems associated with it. As such, one of ordinary skill in the art would be motivated to combine the references, for the reasons set forth in the rejection. Applicant has not pointed out any specific error in the obviousness statement and motivation provided in the rejection.

For these reasons, the rejection of claim 12 is maintained.

Conclusion

14. **THIS ACTION IS MADE FINAL.** Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

15. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Trent J Roche whose telephone number is (703)305-4627. The examiner can normally be reached on Monday - Friday, 9:00 am - 6:30 pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Kakali Chaki can be reached on (703)305-9662. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Trent J Roche
Examiner
Art Unit 2124

TJR

KAKALI CHAKI
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100